Expérimentations autour des architectures d'apprentissage par transfert pour l'extraction de relations biomédicales

Walid Hafiane, Joël Legrand, Yannick Toussaint, Adrien Coulet

Expérimentations autour des architectures d'apprentissage par transfert pour l'extraction de relations biomédicales

Walid Hafiane*, Joël Legrand*, Yannick Toussaint*, Adrien Coulet*,**

*Loria, CNRS, Inria Nancy-Grand Est, Université de Lorraine,
Campus Scientifique, 615 Rue du Jardin-Botanique, 54506 Vandœuvre-lès-Nancy, France
**Inria Paris, Inserm U1138, Université de Paris,
15 rue de l'Ecole de Médecine, 75006 Paris, France
{prénom.nom} @loria.fr

Résumé. L'extraction de relations (ER) consiste à identifier et à structurer automatiquement des relations à partir de textes. Récemment, BERT a permis d'améliorer les performances de plusieurs tâches de TAL, dont l'ER. Cependant, la meilleure façon d'utiliser BERT, dans une architecture d'apprentissage automatique avec une stratégie par transfert reste une question ouverte, car elle dépend à la fois de la tâche et du domaine d'application. Dans ce travail, nous explorons diverses architectures d'ER qui s'appuient sur BERT et deux stratégies de transfert (*gel des poids* ou *réglage fin*) sur deux corpus biomédicaux. Parmi les architectures et stratégies de transfert testées, *BERT-segMCNN avec réglage fin atteint des performances supérieures à l'état de l'art sur les deux corpus (amélioration absolue de 1,73% et 32,77% sur ChemProt et PGxCorpus respectivement). Nos expériences illustrent l'intérêt attendu du réglage fin avec BERT, et de façon plus originale l'intérêt d'ajouter aux représentations de BERT une information structurelle en considérant la segmentation des phrases.

1 Introduction

Le volume de la littérature biomédicale augmente continuellement, ce qui rend les outils de traitement automatique des langues (TAL) attrayants, notamment pour tirer parti des connaissances de domaine qui y sont exprimées. Au sein du TAL, la tâche d'extraction de relations (ER) joue un rôle clé pour l'extraction de connaissances automatique, utiles à des applications telles que les systèmes de questions-réponses, de compréhension du langage naturel ou de résumé automatique de textes.

L'extraction de relations vise à identifier, dans un texte non structuré, toutes les instances d'un ensemble prédéfini de types de relations, entre des entités identifiées (Pawar et al., 2017). Les relations associent alors deux entités nommées, ou plus ; peuvent être typées, orientées et associées à des meta-données. La Figure 1 fournit un exemple de relation orientée, du type "influences", qui associe deux entités (de type Limited-Variation et Pharmacodynamic-phenotype). Nous considérons ici l'extraction de relations binaires et considérons cette extraction comme une tâche de classification qui associe un score à chaque type de relation considéré.

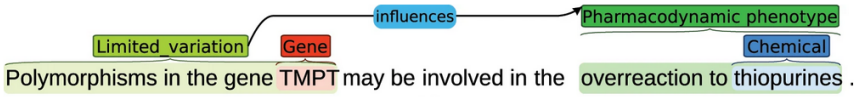

FIG. 1: Exemple de phrase avec quatre entités nommées et une relation entre deux d'entre elles. Cette relation est extraite de PGxCorpus (Legrand et al., 2020).

Dans cet article, nous explorons la tâche d'ER, appliquée au domaine biomédical et du point de vue de l'apprentissage profond.

Pour la plupart des tâches de TAL, les réseaux de neurones profonds ont permis d'améliorer les performances de l'état de l'art et l'ER ne fait pas exception (Collobert et al., 2011; Kumar, 2017). En particulier, les réseaux de neurones convolutifs et récurrents (CNN et RNN pour *Convoluational Neural Network* et *Recurrent Neural Network* en anglais) ont été utilisés avec succès pour les applications biomédicales de cette tâche. D'abord utilisé dans le domaine de la vision, les CNN multicanaux (ou MCNN pour *Multichannel CNN* en anglais) sont des exemples de CNN qui ont été adaptés au TAL pour l'ER (Quan et al., 2016). Les MCNN présente la caractéristique de pouvoir considérer à travers plusieurs canaux, des caractéristiques locales latentes à travers différents vecteurs de plongement lexicaux. BERT (*Bidirectional Encoder Representations from Transformers*) est une autre architecture qui permet de capturer des séquences de mots longues et bidirectionnelles. Pour cela, BERT combine un encodeur à un mécanisme d'attention, et permet d'améliorer les performances de nombreuses tâches de TAL, y compris l'ER (Devlin et al., 2018; Shi et Lin, 2019). Les modèles BERT s'avèrent particulièrement efficaces pour l'apprentissage de certaines caractéristiques du langage, notamment sémantiques. Cependant, ils sont moins performants quand il s'agit de saisir des informations structurelles. En effet, contrairement aux RNN, les modèles BERT ne considèrent par l'ordre dans les séquences d'entrée. Cette information est donc absente des vecteurs de représentation appris par les différentes couches, et la sous-couche d'attention. Cela conduit à une faible contribution des informations structurelles du texte dans les performances de ces modèles. L'unique source de ce type d'information est codé en entrée sous la forme de plongements de positions. Cependant, le signal présent en entrée de BERT se propage à travers les 12 couches dont est constitué BERT, ce qui atténue la représentation de cette information. Li et al. (2019) ont montré dans le cadre de l'ER et d'une architecture de type RNN, qu'un pré-traitement qui consiste à une segmentation des phrases en 5 portions (avant entité1, entité1, entre entités, entité2, après entité 2) renforce l'information structurelle et les performances d'ER. Chen et al. (2020) ont illustré de leur côté l'impact de ce type de pré-traitement avec une architecture de type CNN. A notre connaissance, l'utilisation de ce type de traitement pour l'ER n'a pas été exploré en combinaison avec une architecture de type BERT.

Nous explorons dans ce travail deux stratégies d'apprentissage par transfert pour améliorer les performances des variantes BERT (notées *BERT) sur deux tâches d'ER biomédicales. Nous avons expérimenté des architectures basées sur BERT avec deux stratégies d'apprentissage par transfert (*gel des poids* et *réglage fin*), dans l'hypothèse qu'il pourrait être bénéfique d'ajouter des caractéristiques pertinentes aux vecteurs de représentation de BERT. En particu-

lier, nous explorons la possibilité d'enrichir les modèles BERT avec des informations structurelles en utilisant la segmentation de phrases en post-traitement, ce qui, à notre connaissance, n'a jamais été fait. Les expérimentations sont menées à partir de deux corpus biomédicaux de référence : ChemProt (Kringelum et al., 2016) et PGxCorpus (Legrand et al., 2020).

Cet article est organisé comme suit : la section 2 fournit des éléments de contexte relatifs à notre tâche d'apprentissage (c'est à dire l'ER biomédicales), sur l'architecture BERT et les stratégies d'apprentissage par transfert. La section 3 détaille les architectures basées sur BERT et les stratégies d'apprentissage par transfert mises en œuvre. La section 4 présente les expériences et résultats. La section 5 discute les résultats et conclue.

2 Contexte

2.1 Deux corpus de relations annotées

Nos expériences d'ER s'appuient sur deux corpus de textes biomédicaux en langue anglaise, annotés manuellement, au sein desquels des relations de types distincts sont annotées.

ChemProt contient 10 031 relations entre des protéines et des molécules chimiques, de 13 types différents (Kringelum et al., 2016). Le Tableau 1 présente la répartition par type des relations de l'ensemble d'entraînement. Cette répartition est déséquilibrée puisque le type de relation le plus fréquent représente 39,4% des relations, alors que le moins fréquent n'en représente que 0,1%. Ce dernier est le type "AGONIST INHIBITOR" qui n'est représenté que par 4 exemples. ChemProt est divisé en trois sous-ensembles : apprentissage, validation et test.

PGxCorpus contient 2 875 relations pharmacogénomiques entre facteurs génétiques, médicaments et réponses aux médicaments, de 7 types différents (Legrand et al., 2020). Le Tableau 2 présente la distribution des relations par type. Ici aussi, la répartition est déséquilibrée : le type le plus fréquent est "influences" (32,6%), et le moins fréquent est "causes" (5,8%).

Ces deux corpus ont été choisis pour des raisons distinctes. ChemProt est un corpus qui sert de référence dans de nombreux travaux d'ER et l'apprentissage par transfert, et pour cette raison, il nous permet de comparer nos performances avec l'état de l'art. PGxCorpus quant à lui, est un corpus récent (2020) dédié au domaine de la pharmacogénomique. Il offre la possibilité originale d'étudier l'extraction de relation n-aire et de relations imbriquées.

Intervalle	Type de relation	Taille
$>$1000	INHIBITOR	1642(39,4%)
$>$400	I.-DOWNREGULATOR / SUBSTRATE	487(11,7%) / 480(11,5%)
$>$300	I.-UPREGULATOR / ACTIVATOR	387(9,3%) / 323(7,7%)
$>$200	ANTAGONIST / PRODUCT-OF	235(5,6%) / 233(5,6%)
$>$100	AGONIST / DOWNREGULATOR	156(3,7%) / 131(3,1%)
$>$10	UPREGULATOR / SUBSTRATE-P.	67(1,6%) / 14(0,3%)
$\leq$10	AGONIST-A. / AGONIST-I.	10(0,2%) / 4(0,1%)

TAB. 1: Répartition des relations annotées dans ChemProt, par type.

Intervalle	Type de relation	Taille
>400	Influences / IsAssociatedWith	937(32,6%) / 733(25,5%)
>250	IsEquivalentTo / Decreases	293(10,2%) / 263(9,1%)
>200	Increases / Treats	243(8,5%) / 238(8,3%)
≤200	Causes	168(5,8%)

TAB. 2: Répartition des relations annotées dans PGxCorpus, par type.

2.2 BERT

BERT est la première architecture encodeur qui considère le contexte à la fois en amont et en aval. L'encodeur BERT est constitué d'une pile de 12 couches identiques, où chacune est composée de deux sous-couches. La première est un mécanisme d'auto-attention à plusieurs têtes, et la seconde est de type réseau de neurones à propagation avant, entièrement connecté. Des connexions résiduelles sont utilisées autour de chacune des deux sous-couches, et sont suivies d'une normalisation par couches qui est appliquée après chaque sous-couche. BERT est pré-entraîné automatiquement sur deux tâches (la prédiction de mots masqués, la prédiction de la phrase suivante) sur deux corpus volumineux (Wikipédia, 2,5 milliards de mots ; BookCorpus, 800 millions de mots). Ce pré-entraînement permet à BERT d'acquérir une certaine capacité de généralisation de la langue, avec l'idée de réutiliser cette capacité pour d'autres tâches (l'ER par exemple) et d'autres domaines (le domaine biomédical par exemple).

2.3 L'apprentissage par transfert

D'après Pan et Yang (2010), deux types d'apprentissage par transfert peuvent être distinguées : l'*inductif* et le *transductif*. Le transfert inductif consiste en un transfert d'information entre différentes tâches, généralement au sein d'un même domaine. Le transfert transductif consiste à transférer des informations d'un domaine à l'autre, pour une même tâche.

Suivant l'approche transductive, l'architecture de BERT a été utilisée pour entraîner des modèles adaptés à des domaines spécifiques, comme le domaine de la biomédecine. Cela a donné naissance à plusieurs variantes de BERT, c'est à dire des modèles BERT pré-entraînées sur des corpus différents. BioBERT, par exemple, initialise ses poids avec le modèle pré-entraîné de BERT et poursuit ensuite son entraînement sur un grand ensemble de textes biomédicaux (environ 18 milliards de mots) (Lee et al., 2020). Les performances de BERT et de BioBERT ont été comparées pour la tâche d'ER sur le corpus ChemProt, et montrent la supériorité de BioBERT, illustrant l'intérêt d'affiner le modèle avec des données du domaine. SciBERT est une seconde variante de BERT qui suit la même approche, mais utilise 3,17 milliards de mots de textes scientifiques de domaines divers, dont 82 % relèvent du domaine biomédical (Beltagy et al., 2019). Pour tâche d'ER sur le corpus ChemProt, SciBERT atteint les meilleures performances de l'état de l'art.

Suivant l'approche inductive, les modèles BERT pré-entraînés, de par leur capacité à comprendre et à représenter le langage, sont couramment réutilisés et enrichis (par l'ajout de couches supplémentaires) pour différentes tâches telles que la reconnaissance d'entités nommées, ou l'ER. Dans ce contexte, nous explorons à la fois les stratégies de *gel des poids* et *de*

réglage fin. La stratégie de *gel des poids* consiste à réutiliser les poids de certaines couches d'un modèle précédemment entraîné, et à les "geler", c'est-à-dire ne pas les mettre à jour lors des étapes ultérieures de l'entraînement. On ajoute de nouvelles couches à la suite des couches gelées, de sorte que celles-ci considèrent les sorties des couches gelées comme des entrées pour la tâche d'apprentissage finale. La stratégie de *réglage fin* quant à elle, réutilise également les couches d'un modèle précédemment entraîné, en y ajoutant des couches supplémentaires, mais permet de régler tous les paramètres du modèle relativement à un nouvel ensemble d'apprentissage. Dans le cas de la stratégie de gel des poids, le fait que la rétropropagation du gradient ne s'applique qu'aux neurones ajoutés présente un avantage en termes de temps de calcul. Le réglage fin est en revanche plus gourmand en calcul, mais permet d'obtenir généralement des performances supérieures, en adaptant les paramètres pré-entraînés au nouvel ensemble d'apprentissage.

Dans cette étude, nous explorons l'apprentissage par transfert transductif et inductif pour la tâche d'ER biomédicales. Nous expérimentons diverses variantes de BERT ainsi que les stratégies par transfert de gel des poids et de réglage fin.

3 Méthodes

3.1 Architectures pour l'extraction de relations

3.1.1 Les architectures de l'état de l'art

***BERT+BiLSTM** Pour la stratégie de gel des poids, notre modèle de référence est l'architecture d'ER rapportée dans l'article de SciBERT : un modèle BERT suivi d'un RNN (Beltagy et al., 2019). Par souci de simplicité, nous désignons l'ensemble des modèles BERT (c'est à dire BERT, BioBERT ou SciBERT) par la notation *BERT. Ici *BERT est utilisé comme un extracteur pré-entraîné de plongements de mots contextualisés. Le RNN utilisé est composé d'un BiLSTM à deux couches de taille 200, suivi d'un perceptron multicouche appliqué sur les premier et dernier vecteurs du BiLSTM concaténés. Ce RNN est conçu pour extraire les informations contextuelles de la séquence de plongements lexicaux, afin d'alimenter le classifieur. Nous désignons cette architecture par BERT+BiLSTM, BioBERT+BiLSTM et SciBERT+BiLSTM, ou généralement par *BERT+BiLSTM.

***BERT+MLP** Pour la stratégie de réglage fin, notre modèle de référence est l'architecture utilisée pour l'ER par BERT, BioBERT et SciBERT. Elle consiste en un simple ajout d'une couche linéaire entièrement connectée, en prolongement des modèles BERT pré-entraînés (Devlin et al., 2018; Lee et al., 2020; Beltagy et al., 2019). Nous désignons ces architectures par BERT+MLP, BioBERT+MLP et SciBERT+MLP. Dans l'article de SciBERT, l'ER est évaluée sur ChemProt avec les stratégies de gel des poids (SciBERT+BiLSTM) et de réglage fin (SciBERT+MLP). Nous avons reproduit leurs résultats en vue de les comparer avec d'autres architectures et d'autres corpus.

3.1.2 Architectures proposées

***BERT+MCNN** La première architecture proposée est une extension de BERT avec un MCNN, désignée par *BERT+MCNN. Nous expérimentons cette combinaison, en espérant

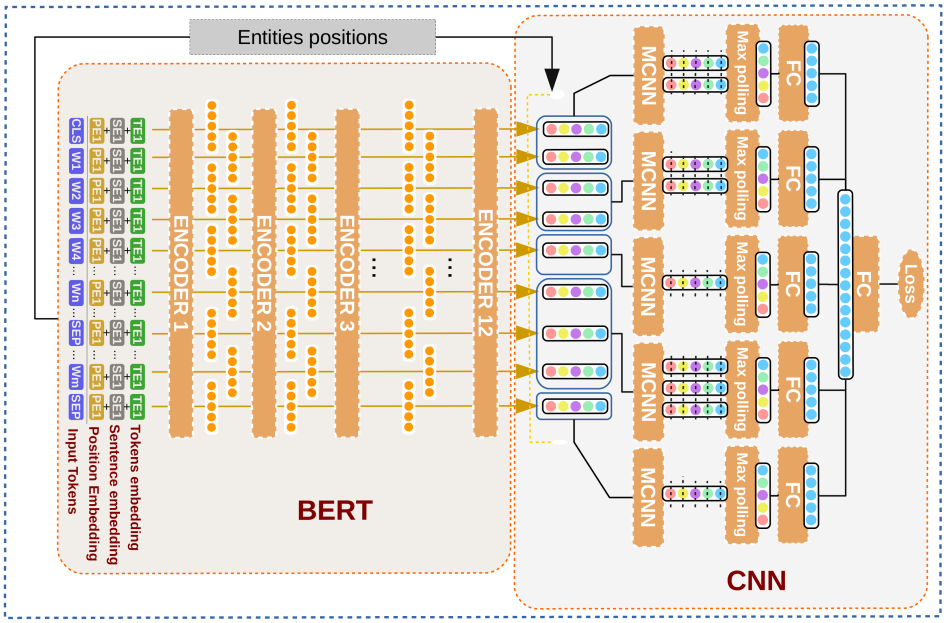

FIG. 2: Représentation schématique de notre architecture BERT-segMCNN.

que le MCNN extraie des informations locales à partir des vecteurs de représentation de BERT. En effet, BERT a la capacité d'extraire des informations contextuelles à longue distance en utilisant un système d'attention, tandis que les informations contextuelles locales capturées par le MCNN peuvent fournir des caractéristiques discriminantes supplémentaires. Le fait que l'ordre entre les différentes dimensions ne soit pas explicite dans les vecteurs de représentation nous amène à les considérer comme des caractéristiques indépendantes et à traiter chaque dimension comme un plongement différent. Par conséquent, chaque dimension représente un canal du MCNN. Cette architecture est suivie d'un max-pooling, qui à son tour alimente un perceptron multicouche. Nous expérimentons cette architecture avec les deux stratégies de transfert.

***BERT+BiLSTM-MCNN** Le deuxième type d'architectures proposé complète BERT avec un BiLSTM et un MCNN. L'exploration de cette architecture est motivée par le fait qu'un BiLSTM peut capturer des caractéristiques sémantiques dans les vecteurs de représentation de BERT alors qu'un MCNN peut capturer les informations locales à partir des mêmes vecteurs. Pour explorer la manière dont ces caractéristiques peuvent être combinées, nous proposons deux architectures : une linéaire (*BERT+BiLSTM-MCNN L.) où un BiLSTM produit les entrées d'un MCNN, suivit d'un maxpooling, puis d'un perceptron multicouche pour la classification ; une parallèle (*BERT+BiLSTM-MCNN P.) où les entrées des MCNN et BLSTM sont les vecteurs de représentation issus de BERT. Les sorties des MCNN et BiLSTM sont concaténées pour alimenter le perceptron multicouche pour la classification. Ces architectures

ne sont testées que dans le cadre de la stratégie de gel des poids, en raison de la profondeur de BERT et de l'extension BiLSTM-MCNN, qui alourdissent les calculs et peuvent conduire à une disparition du gradient.

***BERT+segMCNN** La troisième architecture proposée étend BERT avec cinq MCNN, chacun d'eux prenant comme entrée une partition (ou un segment) différente des vecteurs de représentation issues de BERT. Nous proposons cette architecture pour palier le fait qu'aucun ordre n'est imposé dans les transformer entre les vecteurs de représentation, en particulier dans la couche d'attention. Par conséquent, l'information structurelle peut être limitée dans les vecteurs de représentation de sortie. Afin de renforcer l'architecture *BERT avec une information structurelle, nous avons utilisé cinq MCNN, en parallèle, comme l'illustre la figure 2. Les vecteurs de représentation sont segmentés suivant la partition suivante de la phrase : avant la première entité, la première entité, entre les deux entités, la deuxième entité, et après la deuxième entité. Afin d'effectuer ce post-traitement, les positions des entités sont reportées depuis l'entrée de l'architecture entière jusqu'à l'entrée du bloc segMCNN, comme le montre la figure 2. Une couche de *max pooling* et une couche entièrement connectée sont appliquées après chaque MCNN, les cinq vecteurs résultants sont concaténés, puis cette architecture se termine par un perceptron multicouche pour la classification.

3.2 Conditions Expérimentales

Nous avons définis 4 "fournées" d'expériences, chacune d'elle est associée à un des deux corpus cibles et une des deux stratégies de transfert. Pour chaque fournée nous testons à des fins comparatives les architectures de l'état de l'art à celles ici proposées.

3.2.1 Métriques pour l'évaluation

Selon les expériences, nous avons utilisé comme mesure d'évaluation la *mesure F moyenne macro* ou *micro* (notées F-macro et F-micro). Lorsque l'on souhaite une mesure F unique pour une classification multi-classes, il est possible : de considérer indépendamment chaque exemple, c'est-à-dire de moyenner sur tous les exemples, en les traitant équitablement quelle que soit leur classe (F-micro) ; par une simple moyenne arithmétique sur les classes de traiter toutes les classes équitablement quelle que soit leur taille (F-macro). La F-micro est préférée dans les situations déséquilibrées, et nous la préférons le cas de ChemProt qui comprend des types de relations rares. Nous notons que la F-micro est, sur le plan du calcul, équivalent à l'*accuracy* dans le cas des classes disjointes, ce qui est ici le cas.

3.2.2 Cadre expérimental

Les modèles sont entraînés afin de minimiser la fonction d'entropie croisée. Avec les corpus ChemProt, nos modèles sont entraînés sur l'ensemble d'entraînement, testés sur celui de test, et celui de validation sert à la sélection du modèle et au réglage des hyper-paramètres. Nous avons réutilisé les hyper-paramètres réglés sur ChemProt pour les expériences PGxCorpus. Pour PGxCorpus, il n'y a pas d'ensemble d'entraînement, de validation ou de test prédéfinis. Nous avons donc adopté une stratégie de validation croisée à 10 plis. Les résultats du réglage des hyper-paramètres nous ont fourni divers réglages optimaux pour chaque paire

d'architecture-stratégie de transfert. En conséquence, nous présentons ici les valeurs de paramètres que nous avons utilisés. Pour les CNN, les tailles des filtres de convolution sont de (3, 5, 7) et le nombre de filtres de (3, 6). Pour les BiLSTM, nous avons utilisé deux couches de taille 200, et pour MLP une couche cachée de taille (64, 100). Nous entraînons nos modèles avec une taille de lot de 32 et utilisons deux types de régularisation : un *dropout* de (0,1, 0,25, 0,5) et une régularisation L2 de (0, 0,01). Nous optimisons la fonction de perte en utilisant AdamWithDecay avec une décroissance de (0, 0,01) et un taux d'apprentissage initial de 0,001 pour la stratégie de transfert de gel des poids, et (3.10^{-5} ,5.10^{-5} ,10^{-5}) pour la stratégie de réglage fin. Les nombres d'époques utilisées sont (30, 65) pour la stratégie de gel des poids et (5, 8) pour la stratégie de réglage fin. Les poids sont initialisés par une distribution normale ($\mu = 0, \sigma = 0,02$). La procédure d'apprentissage est initialisée avec différents poids aléatoires (environ 100 fois) et nous rapportons les performances moyennes. Nous indiquons l'écart-type pour analyser la stabilité des modèles.

Les expériences ont été développées en Python avec PyTorch et les versions BERT-base-uncased de BERT, v1.1 de BioBERT-uncased et SciBERT-SciVOCAB-uncased de SciBERT. Le code de nos expériences est disponible à : `https://github.com/hafianewalid/Transfer-Learning-Architectures-for-Biomedical-Relation-Extraction`.

4 Résultats expérimentaux

4.1 Stratégie de gel des poids

	Architecture	F-micro	σ
(Beltagy et al., 2019)	SciBERT+BiLSTM	75,03	–
BERT	+ BiLSTM	64,41	2,42
	+ MCNN	68,26	1,54
	+ BiLSTM-MCNN L.	75,35	1,02
	+ BiLSTM-MCNN P.	62,07	1,33
BioBERT	+ BiLSTM	72,86	1,36
	+ MCNN	77,57	0,70
	+ BiLSTM-MCNN L.	**80,08**	0,80
	+ BiLSTM-MCNN P.	74,85	0,96
SciBERT	+ BiLSTM (Reproduction SOTA)	75,10	1,00
	+ MCNN	77,85	**0,68**
	+ BiLSTM-MCNN L.	79,24	0,76
	+ BiLSTM-MCNN P.	70,45	1,18

TAB. 3: Évaluation des performances de diverses architectures d'apprentissage par transfert pour l'ER sur le corpus ChemProt, en utilisant la stratégie de gel des poids.

	Architecture	Précision	Rappel	F-macro	σ
(Legrand et al., 2020)	MCNN	–	–	45,67	4,51
BERT	+ BiLSTM	54,29	54,82	54,29	0,84
	+ MCNN	57,64	53,84	53,73	1,33
	+ BiLSTM-MCNN L.	71,85	70,47	70,63	1,38
	+ BiLSTM-MCNN P.	54,48	54,37	53,99	**0,28**
BioBERT	+ BiLSTM	57,16	57,32	56,93	0,89
	+ MCNN	69,52	66,25	67,02	2,43
	+ BiLSTM-MCNN L.	**73,05**	71,81	72,09	1,71
	+ BiLSTM-MCNN P.	67,46	64,64	65,39	0,57
SciBERT	+ BiLSTM	62,60	62,26	62,18	1,18
	+ MCNN	69,59	66,57	67,35	1,74
	+ BiLSTM-MCNN L.	72,65	**72,58**	**72,38**	1,04
	+ BiLSTM-MCNN P.	61,54	61,11	61,00	1,45

TAB. 4: Évaluation des performances de diverses architectures d'apprentissage par transfert pour l'ER sur le corpus PGxCorpus, en utilisant la stratégie de gel des poids.

Avec ChemProt Les résultats obtenus avec la stratégie de gel des poids sur le corpus ChemProt sont présentés dans le Tableau 3. La première ligne du tableau présente les résultats de Beltagy et al. (2019), que nous avons reproduits et présentés sur la ligne 10 du tableau (dénommée reproduction SOTA). Elle présente un écart de 0,07 % de F-micro (écart-type = 1) par rapport à la performance rapportée par Beltagy et al. (2019). Quelle que soit la variante de BERT que nous avons utilisée, *BERT+BiLSTM-MCNN L. produit les modèles les plus performants sur le corpus ChemProt. BioBERT+BiLSTM-MCNN obtient les meilleures performances avec 80,08 % de F-micro.

Avec PGxCorpus Les résultats obtenus avec la stratégie de gel des poids sur PGxCorpus sont présentés dans le Tableau 3. La première ligne présente les résultats de citeLegrand2020, obtenus avec un simple MCNN, sans modèle BERT pré-entraîné. Comme attendu, nous constatons que toutes nos architectures (toutes basées sur BERT) dépassent le modèle de référence tout en étant plus stables. En particulier, nous constatons qu'indépendamment de la variante de BERT utilisée, l'architecture *BERT-BiLSTM-MCNN L. produit les modèles les plus performants sur PGxCorpus, de manière similaire à ce que nous avons observé avec ChemProt. Nous notons également que, cette fois-ci, SciBERT+BiLST-M-MCNN L. dépasse légèrement la version BioBERT de la même architecture.

4.2 Stratégie de réglage fin

Avec ChemProt Les résultats obtenus avec la stratégie de réglage fin sur ChemProt sont présentés dans le Tableau 5. La première ligne du Tableau 5 indique les résultats de Beltagy et al. (2019), que nous avons reproduits et présentés sur la 8ème ligne du tableau (appelée reproduction SOTA). Elle présente un écart de 1,2% de F-micro (écart-type = 1,47) par rapport à la performance rapportée dans

Expérimentations autour des architectures d'apprentissage par transfert

	Architecture	**F-micro**	σ
(Beltagy et al., 2019)	SciBERT+MLP	83,64	–
BERT	+ MLP	79,28	1,21
	+ MCNN	72,18	1,73
	+ segMCNN	81,43	0,91
BioBERT	+ MLP	82,28	3,27
	+ MCNN	83,90	1,11
	+ segMCNN	**85,37**	**0,67**
SciBERT	+ MLP (Reproduction SOTA)	82,44	1,47
	+ MCNN	82,98	0,96
	+ segMCNN	84,77	**0,67**

TAB. 5: Évaluation des performances de diverses architectures d'apprentissage par transfert pour l'ER sur le corpus ChemProt, en utilisant la stratégie de réglage fin.

	Architecture	**Précision**	**Rappel**	**F-macro**	σ
(Legrand et al., 2020)	MCNN	–	–	45,67	4,51
BERT	+ MLP	70,80	69,70	69,88	4,03
	+ MCNN	71,73	73,39	72,22	**0,98**
	+ segMCNN	74,31	74,53	74,17	1,08
BioBERT	+ MLP	73,49	68,84	70,47	5,87
	+ MCNN	74,40	71,16	72,23	4,30
	+ segMCNN	77,38	77,24	77,00	2,56
SciBERT	+ MLP	75,61	75,44	75,32	2,11
	+ MCNN	75,88	76,08	75,70	1,04
	+ segMCNN	**78,15**	**79,17**	**78,44**	1,07

TAB. 6: Évaluation des performances de diverses architectures d'apprentissage par transfert pour l'ER sur le corpus PGxCorpus, en utilisant la stratégie de réglage fin.

Beltagy et al. (2019). On observe que pour chaque variante de BERT, les modèles obtenus avec l'architecture *BERT+segMCNN produisent les meilleurs performances. En particulier, nous constatons que *BERT+segMCNN nous a permis d'atteindre un niveau de performance légèrement supérieur au niveau de l'état de l'art (85,37%, amélioration absolue de 1,73% par rapport à Beltagy et al. (2019)). Nous observons également que l'ajout de notre approche de segmentation de phrase, associée aux 5 MCNN en parallèle (segMCNN) surpasse les résultats de l'état de l'art de BioBERT et SciBERT.

Avec PGxCorpus Les résultats obtenus avec la stratégie de réglage fin sur PGxCorpus sont présentés dans le Tableau 6. La première ligne présente les résultats de Legrand et al. (2020), obtenus avec un simple MCNN, sans modèle BERT pré-entraîné. Comme attendu, nous constatons que toutes nos architectures (toutes basées sur BERT) surpassent les modèles de référence et sont plus stables. On observe que quelque soit la variante de BERT considérée, le modèle *BERT+segMCNN produit les meilleurs performances. Nous notons en particulier qu'avec SciBERT+segMCNN nous établissont une nouvelle performance de référence pour l'état de l'art de l'ER pharmacogénomique avec PGxCorpus (78,44%, amélioration absolue de 32,77%). Il semble qu'une première amélioration est obtenue avec l'architecture *BERT+MCNN, qui est encore améliorée avec *BERT+segMCNN. On note également que, contrairement aux résultats obtenus sur ChemProt, la meilleure variante de BERT est SciBERT (*vs.* BioBERT sur ChemProt).

De façon générale les expériences avec les architectures *BERT+MCNN ont été réalisées à la fois avec les deux stratégies de transfert, gel des poids et réglage fin. Elles illustrent que cette dernière produit de bien meilleures performances, ce qui pouvait être attendu.

5 Discussion et Conclusion

Dans le cadre de l'ER, nous avons formulé l'hypothèse que les représentations de langage apprises avec BERT pourraient être enrichies avec des éléments d'information structurelle, notamment par la considération de la phrase sous forme de plusieurs segments, ainsi que par l'utilisation de l'information locale contenue dans ses vecteurs de représentation. Nos résultats illustrent tout d'abord le fait que la variante de BERT utilisée a une incidence sur les performances finales de la tâche d'ER. De plus, cet impact varie en fonction du corpus cible et du types spécifique de relations associées. Nous observons également, de façon attendue, que les stratégies de réglage fin, même si elles sont coûteuses en termes de calcul, produisent de meilleures performances.

Pour conclure, nous avons réalisé des expériences avec plusieurs architectures basées sur BERT et plusieurs stratégies de transfert pour la tâche d'ER, sur deux corpus biomédicaux. Notre choix d'architectures a été motivé par la volonté d'exploiter plusieurs caractéristiques du langage naturel : des caractéristiques locales extraites par le MCNN, des caractéristiques contextuelles grâce au BiLSTM, et des caractéristiques structurelles provenant d'une approche de segmentation de phrases. Nous avons proposé différentes architectures adaptées aux stratégies de gel des poids et de réglage fin (*BERT+BiLSTM-MCNN L. et *BERT+segMCNN, respectivement) conduisant à une amélioration des performances au niveau de l'état de l'art pour nos tâches spécifiques d'ER biomédical. Même si notre contribution empirique se limite à un sous-ensemble de variantes de BERT et à la tâche spécifique d'ER biomédical, nous pensons que nos architectures (en particulier *BERT+BiLSTM-MCNN L. et *BERT+segMCNN) peuvent convenir à d'autres variantes de BERT, ainsi qu'à d'autres tâches et domaines.

Références

Beltagy, I., K. Lo, et A. Cohan (2019). Scibert : A pretrained language model for scientific text. *EMNLP*.

Chen, Y., K. Wang, W. Yang, Y. Qin, R. Huang, et P. Chen (2020). A multi-channel deep neural network for relation extraction. *IEEE Access 8*, 13195–13203.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, et P. P. Kuksa (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res. 12*, 2493–2537.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2018). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT* (Mlm).

Kringelum, J., S. K. Kjærulff, S. Brunak, O. Lund, T. I. Oprea, et O. Taboureau (2016). Chemprot-3.0 : a global chemical biology diseases mapping. *Database J. Biol. Databases Curation 2016*.

Kumar, S. (2017). A survey of deep learning methods for relation extraction. *CoRR abs/1705.03645*.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, et J. Kang (2020). Biobert : a pre-trained biomedical language representation model for biomedical text mining. *Bioinform. 36*(4), 1234–1240.

Legrand, J., R. Gogdemir, C. Bousquet, K. Dalleau, M. D. Devignes, W. Digan, C. J. Lee, N. C. Ndiaye, N. Petitpain, P. Ringot, M. Smaïl-Tabbone, Y. Toussaint, et A. Coulet (2020). PGxCorpus, a manually annotated corpus for pharmacogenomics. *Scientific Data 7*(1), 1–13.

Li, Z., J. Yang, X. Gou, et X. Qi (2019). Recurrent neural networks with segment attention and entity description for relation extraction from clinical texts. *Artif. Intell. Medicine 97*, 9–18.

Pan, S. J. et Q. Yang (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng. 22*(10), 1345–1359.

Pawar, S., G. K. Palshikar, et P. Bhattacharyya (2017). Relation extraction : A survey. *CoRR abs/1712.05191*.

Quan, C., L. Hua, X. Sun, et W. Bai (2016). Multichannel convolutional neural network for biological relation extraction. *BioMed Research International*.

Shi, P. et J. Lin (2019). Simple bert models for relation extraction and semantic role labeling. *CoRR abs/1904.05255*.

Summary

Relation extraction (RE) consists in identifying and structuring automatically relations of interest from texts. Recently, BERT improved the top performances for several NLP tasks, including RE. However, the best way to use BERT, within a machine learning architecture, and within a transfer learning strategy is still an open question since it is highly dependent on each specific task and domain. Here, we explore various BERT-based architectures and transfer learning strategies (i.e., *frozen* or *fine-tuned*) for the task of biomedical RE on two corpora. Among tested architectures and strategies, our *BERT-segMCNN with fine-tuning reaches performances higher than the state-of-the-art on the two corpora (1.73 % and 32.77 % absolute improvement on ChemProt and PGxCorpus corpora respectively). More generally, our experiments illustrate the expected interest of fine-tuning with BERT, but also the unexplored advantage of using structural information (with sentence segmentation), in addition to the context classically leveraged by BERT.